Improved backward error bounds for LU and Cholesky factorizations

Siegfried M. Rump, Claude-Pierre Jeannerod

IMPROVED BACKWARD ERROR BOUNDS FOR LU AND CHOLESKY FACTORIZATIONS

SIEGFRIED M. RUMP* AND CLAUDE-PIERRE JEANNEROD†

Abstract. Assuming standard floating-point arithmetic (in base β, precision p) and barring underflow and overflow, classical rounding error analysis of the LU or Cholesky factorization of an $n \times n$ matrix A provides backward error bounds of the form $|\Delta A| \leqslant \gamma_n |\widehat{L}||\widehat{U}|$ or $|\Delta A| \leqslant \gamma_{n+1}|\widehat{R}^T||\widehat{R}|$. Here, $\widehat{L}$, $\widehat{U}$, and $\widehat{R}$ denote the computed factors, and γ_n is the usual fraction $nu/(1-nu) = nu+\mathcal{O}(u^2)$ with u the unit roundoff. Similarly, when solving an $n \times n$ triangular system $Tx = b$ by substitution, the computed solution $\widehat{x}$ satisfies $(T + \Delta T)\widehat{x} = b$ with $|\Delta T| \leqslant \gamma_n |T|$. All these error bounds contain quadratic terms in u and limit n to satisfy either $nu < 1$ or $(n+1)u < 1$. We show in this paper that the constants γ_n and γ_{n+1} can be replaced by nu and $(n+1)u$, respectively, and that the restrictions on n can be removed.

To get these new bounds the main ingredient is a general framework for bounding expressions of the form $|\rho - s|$, where s is the exact sum of a floating-point number and $n - 1$ real numbers, and where ρ is a real number approximating the computed sum $\widehat{s}$. By instantiating this framework with suitable values of ρ, we obtain improved versions of the well-known Lemma 8.4 from [N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, 2002] (used for analyzing triangular system solving and LU factorization) and of its Cholesky variant. All our results hold for rounding to nearest with any tie-breaking strategy and no matter what the order of summation.

Key words. floating-point summation, rounding error analysis, unit in the first place, backward error, LU factorization, Cholesky factorization, triangular system solving

AMS subject classifications. 65G50, 65F05 **Date:** January 8, 2014

1. Introduction.

Let $\mathbb{F}$ denote a standard set of floating-point numbers in radix β and precision p. Assuming standard floating-point arithmetic with rounding to nearest and barring underflow and overflow, our goal is to show that the following backward error bounds hold for LU and Cholesky factorizations and triangular system solving: if for some $A \in \mathbb{F}^{m\times n}$ with $m \geqslant n$ Gaussian elimination runs to completion, then the computed factors $\widehat{L}$ and $\widehat{U}$ satisfy

$$\widehat{L}\widehat{U} = A + \Delta A, \qquad |\Delta A| \leqslant nu|\widehat{L}||\widehat{U}|, \tag{1.1a}$$

with u the unit roundoff; similarly, if for some symmetric $A \in \mathbb{F}^{n\times n}$ the Cholesky decomposition runs to completion, then the computed factor $\widehat{R}$ satisfies

$$\widehat{R}^T\widehat{R} = A + \Delta A, \qquad |\Delta A| \leqslant (n+1)u|\widehat{R}^T||\widehat{R}|; \tag{1.1b}$$

furthermore, if $Tx = b$ is solved by substitution for $b \in \mathbb{F}^n$ and nonsingular triangular $T \in \mathbb{F}^{n\times n}$, then the computed solution $\widehat{x}$ satisfies

$$(T + \Delta T)\widehat{x} = b, \qquad |\Delta T| \leqslant nu|T|. \tag{1.1c}$$

Each of these bounds improves upon the corresponding classical one, that is,

$$\gamma_n|\widehat{L}||\widehat{U}|, \quad \gamma_{n+1}|\widehat{R}^T||\widehat{R}|, \quad \gamma_n|T|, \tag{1.2}$$

*Institute for Reliable Computing, Hamburg University of Technology, Schwarzenbergstraße 95, Hamburg 21071, Germany, and Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan (rump@tuhh.de).

†INRIA, Laboratoire LIP (CNRS, ENS de Lyon, INRIA, UCBL), Université de Lyon, 46 allée d'Italie 69364 Lyon cedex 07, France (claude-pierre.jeannerod@ens-lyon.fr).

respectively, with γ_n the usual fraction $\gamma_n := nu/(1-nu) = nu + \mathcal{O}(u^2)$; see [3, Theorems 9.3, 10.3, 8.5]. Since by definition γ_n contains quadratic terms in u and assumes implicitly that $nu < 1$, in each case our improvement is twofold in the sense that both the $\mathcal{O}(u^2)$ terms and the restriction on n are removed.

Of course, from the point of view of a priori backward error analysis, the classical bounds (1.2) are already entirely satisfactory and, as Wilkinson [11] puts it, "The precise error bound is not of great importance." Nevertheless, it is worth noting that such bounds, which have been available since over 50 years, can be replaced by ones that are both simpler and more general.

Our results extend the recent ones in [7, 5], where it was shown that the classical terms γ_{n-1} and γ_n in the error bounds of floating-point sums and inner products can be replaced by $\mathcal{O}(u^2)$-free and unconditional terms $(n-1)u$ and nu. In particular, the inner product analysis in [5] implies immediately that the classical error bound for matrix multiplication with inner dimension n simplifies to $|\widehat{C} - AB| \leqslant nu|A||B|$ without restriction on n. This analysis, however, does not suffice to improve the classical bounds (1.2) into the new ones in (1.1): for LU and Cholesky factorization and triangular system solving, what will be required is a refinement of (a variant of) the well-known Lemma 8.4 from [3] via some careful rounding error analysis of sums of one floating-point number and $n-1$ real numbers.

Note that every result, from summation to inner product to the higher-order matrix algorithms considered here, is based on an individual analysis. Although each analysis covers a family of algorithms (by allowing any ordering in the sums), there seems to be no panacea to generally remove $\mathcal{O}(u^2)$ terms. Indeed examples exist where γ_n cannot be replaced by nu (see the end of Section 5).

Before presenting our approach and contributions in detail, we show how to treat all backward error analyses underlying the well-known bounds (1.2) in a uniform way. Classical algorithms for triangularizing $n \times n$ matrices or solving $n \times n$ triangular systems consist of repeatedly evaluating expressions y of the form

$$y \in \left\{s, s/b_k, \sqrt{s}\right\}, \qquad s = c - \sum_{i=1}^{k-1} a_i b_i, \qquad k \leqslant n, \tag{1.3a}$$

for some scalars a_i, b_i, c. Specifically, the patterns s and s/b_k appear in Gaussian elimination for LU factorization or when solving triangular linear systems by substitution, while s/b_k and $\sqrt{s}$ are the expressions typically used by Cholesky factorization. Consequently, to analyze the behavior of such algorithms in floating-point arithmetic it suffices to bound the rounding errors committed during the evaluation of y in (1.3a). Assume that

$$c \text{ and all the } a_i \text{ and } b_i \text{ are in } \mathbb{F} \tag{1.3b}$$

and let $\widehat{y}$ be the result of a floating-point evaluation of y. For such $\widehat{y}$, backward error results are given for example in Higham's book [3], where Lemma 8.4 covers the case $y \in \left\{s, s/b_k\right\}$ and Problem 10.3 considers the case $y = \sqrt{s}$. Writing θ_ℓ for reals such that

$$|\theta_\ell| \leqslant \gamma_\ell = \frac{\ell u}{1-\ell u} \qquad \text{if } \ell u < 1, \tag{1.4}$$

these backward error results can be put in concise form as follows.[1]

[1]Hereafter, superscripts are used to indicate that the $\theta_\ell^{(i)}$ may be pairwise distinct and also

If $y = s$ then

$$\widehat{y}(1+\theta_{k-1}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_{k-1}^{(i)}). \tag{1.5a}$$

If $y = s/b_k$ then

$$b_k\widehat{y}(1+\theta_k) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_k^{(i)}). \tag{1.5b}$$

If $y = \sqrt{s}$ then

$$\widehat{y}^2(1+\theta_{k+1}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_{k+1}^{(i)}). \tag{1.5c}$$

Each of the three identities above holds no matter in what order of evaluation $\widehat{y}$ was produced, provided that no underflow or overflow occurred. Formally, *no matter what the order of evaluation* means to compute the sum using any of the rooted binary trees whose k leaves are c and the $a_i b_i$, and whose $k-1$ inner nodes have degree two and contain one intermediate sum each. For example, when $k = 4$ amongst the possible schemes defined by such summation trees are

$$\big((c - a_1b_1) - a_2b_2\big) - a_3b_3, \quad (c - a_1b_1) - (a_2b_2 + a_3b_3), \quad c - \big((a_1b_1 + a_2b_2) + a_3b_3\big).$$

Note also that for each of the equations in (1.5) the scalar c is kept unperturbed. This makes it possible to obtain the backward error bounds (1.2): for the LU or Cholesky decomposition, c plays the role of a given entry of the input matrix, and when solving $Tx = b$ by substitution, c is an entry of the right-hand side b.

All identities in (1.5) are of the form

$$\rho(1+\theta_{k+\ell-1}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_{k+\ell-1}^{(i)})$$

for suitable values of ρ and ℓ: depending on the expression taken by y in (1.3a), the real number ρ is either $\widehat{y}$, $b_k\widehat{y}$, or $\widehat{y}^2$, and the integer ℓ is either 0, 1, or 2. In any case, ρ is (an approximation of) the computed value of the sum s as in (1.3). Furthermore, due to (1.4) this generic backward error result is equivalent to the forward error bound

$$|\rho - s| \leqslant \gamma_{k+\ell-1}\Big(|\rho| + \sum_{i=1}^{k-1} |a_i b_i|\Big). \tag{1.6}$$

Note in particular that this bound contains a quadratic term in u and holds only if the condition $(k+\ell-1)u < 1$ is satisfied.

---

distinct from θ_ℓ. In addition, the terms $\theta_k^{(i)}$ and $\theta_{k+1}^{(i)}$ are not best possible and can both be replaced by $\theta_{k-1}^{(i)}$. This is easily deduced from Higham's analysis in [3, pp. 141, 553] but does not impact the resulting backward error bounds for triangular system solving and LU and Cholesky factorizations; the reason is that such bounds are governed by the terms θ_k and θ_{k+1} in the left-hand sides of (1.5b) and (1.5c).

In this paper we show that the constant $\gamma_{k+\ell-1}$ in (1.6) can be replaced by the unconditional and $\mathcal{O}(u^2)$-free value $(k+\ell-1)u$. This proves the three bounds in (1.1).

To do so, we introduce a more general result. Instead of considering only special inner products $c-\sum_{i=1}^{k-1} a_i b_i$ as in (1.3), we allow the $a_i b_i$ to be replaced by $k-1 \leqslant n-1$ arbitrary real numbers. In other words, we consider general sums of the form

$$s = x_1 + \cdots + x_n, \qquad x_i \in \mathbb{R} \text{ for all } i \neq j \text{ and } x_j \in \mathbb{F} \text{ for some fixed } j. \tag{1.7}$$

Writing $\mathrm{fl}(x_i)$ to denote an element of $\mathbb{F}$ nearest to x_i, let $\widehat{s}$ be the result of the sum $\mathrm{fl}(x_1)+\cdots+\mathrm{fl}(x_n)$ obtained in floating-point arithmetic, with rounding to nearest and no matter what the tie-breaking rule and what the order of evaluation. Furthermore, let ρ be a real number approximating $\widehat{s}$ as

$$|\rho - \widehat{s}| \leqslant \ell u |\rho| \quad \text{for some } \ell \in \mathbb{N}.$$

Then, we show that in the absence of underflow and overflow,

$$|\rho - s| \leqslant (n + \ell - 1)u\Big(|\rho| + \sum_{i=1, i\neq j}^{n} |x_i|\Big). \tag{1.8}$$

This result holds without any restriction on n and does not involve quadratic terms in u.

By specializing (1.8) with suitable values of ρ and ℓ, we also obtain the following improved versions of the classical backward error results (1.5) and (1.2): First, for each θ_ℓ (and $\theta_\ell^{(i)}$) appearing in (1.5) we can take

$$|\theta_\ell| \leqslant \ell u \quad \text{for all } \ell,$$

instead of $|\theta_\ell| \leqslant \gamma_\ell$ for $\ell u < 1$ as in (1.4). In other words, Higham's Lemma 8.4 and its Cholesky variant hold with the γ_ℓ terms replaced by ℓu and with no restriction on ℓ. Second, using these improved versions of the bounds in (1.5) we show that for LU and Cholesky factorizations and triangular system solving, the upper bounds in (1.2) can be replaced by those in (1.1). As long as neither underflow nor overflow occurs, each of these new backward error bounds holds in standard floating-point arithmetic with rounding to nearest and for any radix β, any tie-breaking rule, any dimension, and any evaluation order of the sums involved.

The rest of this paper is organized as follows. Section 2 provides the necessary definitions and properties about floating-point arithmetic as well as two recent error bounds on floating-point summation. The proof of the bound (1.8) is then given in Section 3. In this section we also present some specializations of this bound that will be used later to refine (1.5). Furthermore, in the particular situation where $(\rho, \ell) = (\widehat{s}, 0)$ and all the x_i are in $\mathbb{F}$ and accumulated recursively one after the other, we remark that we also have

$$|\widehat{s} - s| \leqslant (n-1)u\Big(|\widehat{s}| + \sum_{i=3}^{n} |x_i|\Big).$$

Applications of (1.8) are then detailed in Section 4. We start with our improved version of Higham's Lemma 8.4 and its direct application to triangular system solving, and then turn to the improved backward error bounds for LU and Cholesky factorizations. Some concluding remarks in Section 5 finish the paper.

2. Preliminaries.

2.1. Floating-point numbers and rounding to nearest. Throughout this paper $\mathbb{F}$ denotes a set of finite floating-point numbers similar to those defined in the IEEE 754-2008 standard [4], with base β, precision p, and exponent range $[e_{\min}, e_{\max}]$. In particular, $\beta \geqslant 2$, $p \geqslant 1$, and $\mathbb{F}$ is symmetric and contains zero. (If $p = 1$ then every element of $\mathbb{F}$ is either zero or an integer power of the base.)

A nonzero number in $\mathbb{F}$ is *normal* if its magnitude is at least $\beta^{e_{\min}}$, and *subnormal* otherwise. Accordingly, writing Ω for the largest number in $\mathbb{F}$, we say that a real number t lies in the *normal range* of $\mathbb{F}$ if $\beta^{e_{\min}} \leqslant |t| \leqslant \Omega$, and that it lies in the *subnormal range* of $\mathbb{F}$ if $0 < |t| < \beta^{e_{\min}}$.

We associate with the set $\mathbb{F}$ a round-to-nearest function fl, which can be any map from $\mathbb{R}$ to $\mathbb{F} \cup \{\pm\infty\}$ such that for all $t \in \mathbb{R}$ and if no overflow occurs,

$$|\mathrm{fl}(t) - t| = \min_{s \in \mathbb{F}} |s - t|.$$

Consequently, no assumption is made on the way of breaking ties.

Here and hereafter, overflow is defined in the same way as in the IEEE 754-2008 standard [4, §7.4]. Concerning underflow, we follow Kahan in [6]: when rounding a real number t, we say that *underflow* occurs if t is in the subnormal range of $\mathbb{F}$ without being in $\mathbb{F}$, that is, if

$$0 < |t| < \beta^{e_{\min}} \quad \text{and} \quad \mathrm{fl}(t) \neq t.$$

In terms of the IEEE 754-2008 standard this corresponds precisely to the event when the underflow flag is raised, assuming default exception handling; see [4, §7.5]. With this definition floating-point addition cannot cause underflow, since the sum of two floating-point numbers is exact when it lies in the subnormal range of $\mathbb{F}$; see for example [2]. Concerning other operations, underflow or overflow can occur for multiplication and division, but not for square root.

2.2. Basic properties. Assuming $\mathbb{F}$ and fl as above, we have the following well-known properties. First, if neither underflow nor overflow occurs when rounding $t \in \mathbb{R}$ to $\mathrm{fl}(t)$, then the errors relative to t and $\mathrm{fl}(t)$ are both bounded by the unit roundoff $u = \frac{1}{2}\beta^{1-p}$. In other words,

$$\mathrm{fl}(t) = t(1 + \epsilon_1), \qquad |\epsilon_1| \leqslant u, \tag{2.1a}$$

$$= \frac{t}{1 + \epsilon_2}, \qquad |\epsilon_2| \leqslant u. \tag{2.1b}$$

Note that since β and p are positive integers, we always have $u \leqslant 1/2$. The relation in (2.1a) is commonly referred to as the *standard model* of floating-point arithmetic. It is used in [3] together with the variant (2.1b) to derive the classical backward error results we have mentioned in introduction.

Second, for two floating-point numbers a and b it is known that $|\mathrm{fl}(a+b)-(a+b)| \leqslant \min\{|a|, |b|\}$ if no overflow occurs; see for example [9] as well as [3, p. 91]. By combining this bound with (2.1b) and the fact that $\mathrm{fl}(a + b)$ cannot cause underflow, we deduce the following: if $a, b \in \mathbb{F}$, then

$$|\mathrm{fl}(a + b) - (a + b)| \leqslant \min\{u|\mathrm{fl}(a + b)|, |a|, |b|\} \quad \text{in the absence of overflow.} \tag{2.2}$$

A third set of properties is obtained by considering the notion of *unit in the first place* (ufp), which was introduced in [8] and provides refinements to (2.1): a real number t being given, we have $\mathrm{ufp}(0) = 0$ and, if $t \neq 0$, $\mathrm{ufp}(t) = \beta^{\lfloor \log_\beta |t| \rfloor}$. Hence

$$\mathrm{ufp}(t) \leqslant |t| \tag{2.3a}$$

for all t, and if t is nonzero then its ufp can be thought of as the weight of its first nonzero digit in base-β representation. The functions ufp and fl are known to be related as follows (see for example (2.13) and (2.18) in [8]). On the one hand, for $t \in \mathbb{R}$ and in the absence of overflow,

$$|\mathrm{fl}(t)| = (1 + k \cdot 2u)\,\mathrm{ufp}(t) \quad \text{for some } k \in \mathbb{N}. \tag{2.3b}$$

This implies in particular that

$$\text{either} \quad \mathrm{ufp}(t) = |\mathrm{fl}(t)| \quad \text{or} \quad (1+2u)\mathrm{ufp}(t) \leqslant |\mathrm{fl}(t)|. \tag{2.3c}$$

On the other hand, in the absence of both underflow and overflow,

$$|\mathrm{fl}(t) - t| \leqslant u \cdot \mathrm{ufp}(t). \tag{2.3d}$$

Hence $|\epsilon_1|$ and $|\epsilon_2|$ in (2.1) admit the sharper bounds $u \cdot \mathrm{ufp}(t)/|t|$ and $u \cdot \mathrm{ufp}(t)/|\mathrm{fl}(t)|$, respectively.

2.3. Previous results: a priori error bounds for floating-point sums. When using floating-point arithmetic as just described to evaluate $s = x_1 + \cdots + x_n$, two types of bounds on $|\widehat{s} - s|$ have been obtained recently. They hold for all n and no matter what order of evaluation was used to produce the result $\widehat{s}$. First, if all the x_i are in $\mathbb{F}$, then it was shown in [5, §3] that in the absence of overflow

$$\left|\widehat{s} - \sum_{i=1}^{n} x_i\right| \leqslant (n-1)u \sum_{i=1}^{n} |x_i|.$$

This bound had already appeared in [7] in the special case of *recursive summation*, that is, when $\widehat{s}$ is obtained using the evaluation order $(\ldots((x_1 + x_2) + x_3) + \cdots) + x_n$. Second, if all the x_i are in $\mathbb{R}$ and such that $\mathrm{fl}(x_i)$ does not underflow, then it was shown in [5, §4] that in the absence of overflow

$$\left|\widehat{s} - \sum_{i=1}^{n} x_i\right| \leqslant nu \sum_{i=1}^{n} |x_i|. \tag{2.4}$$

Here, $\widehat{s}$ is obtained by rounding each x_i to $\mathrm{fl}(x_i)$ and then adding all the $\mathrm{fl}(x_i)$ in any given order. By taking each x_i to be of the form $a_i b_i$ with a_i, b_i in $\mathbb{F}$, we see that this second bound covers in particular the case of inner products of floating-point vectors of length n.

The above bounds thus improve upon the classical ones given in [3, pp. 63, 82] for sums of floating-point numbers and inner products of floating-point vectors: the terms $(n-1)u$ and nu replace the classical terms γ_{n-1} and γ_n, and the restrictions on n are removed.

In this paper we will need (2.1b) but also each of the results in (2.2), (2.3), and (2.4). Specifically, (2.2) and (2.4) are used in the proof of Theorem 3.1, while the ufp-based properties given in (2.3) are used to establish the second part of Corollary 3.2.

3. Main results. We start by showing that if some x_j is a floating-point number, then $|x_j|$ in the right-hand side of (2.4) can be replaced by $|\widehat{s}|$ and the term nu by $(n-1)u$. The difficulty is that $|\widehat{s}|$ may be small, even zero. Moreover, we show that the computed sum $\widehat{s}$ can be replaced by some real number when increasing the constant $n-1$ appropriately.

THEOREM 3.1. *Given $n \in \mathbb{N}_{>0}$ and $j \in \{1,\dots,n\}$, let $x_1,\dots,x_n \in \mathbb{R}$ be such that $x_j \in \mathbb{F}$ and, for all $i \neq j$, $\mathrm{fl}(x_i)$ does not underflow. Let $\widehat{s}$ be a floating-point sum of $\mathrm{fl}(x_1),\dots,\mathrm{fl}(x_n)$ no matter what the order of evaluation, and let $\rho \in \mathbb{R}$ be such that*

$$|\rho - \widehat{s}| \leqslant \ell u|\rho| \quad \text{for some } \ell \in \mathbb{N}.$$

Then, in the absence of overflow,

$$\Delta := \rho - \sum_{i=1}^{n} x_i \qquad \textit{satisfies} \qquad |\Delta| \leqslant (n+\ell-1)u\Big(|\rho| + \sum_{i=1,i\neq j}^{n} |x_i|\Big).$$

Proof. The proof is by induction on n. For $n=1$ the assertion $|\rho - x_1| \leqslant \ell u|\rho|$ is true because $x_1 \in \mathbb{F}$ implies $\widehat{s} = x_1$. For $n \geqslant 2$ we assume that the result is true up to $n-1$, and we fix one evaluation order of the summation. This evaluation order defines one specific arithmetic expression, which in turn is represented by a rooted binary tree as explained in the introduction. In this summation tree, let $s_1 \in \mathbb{R}$ be the node where $x_j \in \mathbb{F}$ is added and let $\widehat{s}_1 \in \mathbb{F}$ denote its rounded value, that is,

$$\widehat{s}_1 = \mathrm{fl}(s_1), \qquad s_1 = x_j + \widehat{s}_2. \tag{3.1}$$

Here $\widehat{s}_2$ is the root of a summation tree adding the elements of $\{x_i : i \in I_2\}$ for some non-empty index set $I_2 \subseteq \{1,\dots,n\}\backslash\{j\}$. Define $I_1 = \{1,\dots,n\}\backslash I_2$ and $I_1' = I_1\backslash\{j\}$, and let n_1 and n_2 be the cardinalities of I_1 and I_2, respectively. In particular, $1 \leqslant n_1, n_2 \leqslant n-1$ and $n_1 + n_2 = n$. Furthermore, $\widehat{s}$ is by definition the root of a summation tree adding $x_1,\dots,x_n$, but due to (3.1) it is also the root of a summation tree $\mathcal{T}$ adding the $n_1 < n$ elements of $\{\widehat{s}_1\} \cup \{x_i : i \in I_1'\}$. Abbreviating

$$\Delta_1 = \rho - (\widehat{s}_1 + \sum_{i\in I_1'} x_i) \qquad \text{and} \qquad \Delta_2 = \widehat{s}_2 - \sum_{i\in I_2} x_i, \tag{3.2}$$

we use (3.1) to write $\Delta = \Delta_1 + \widehat{s}_1 - s_1 + \Delta_2$, so that

$$|\Delta| \leqslant |\Delta_1| + |\widehat{s}_1 - s_1| + |\Delta_2|. \tag{3.3}$$

Since $\widehat{s}_1 \in \mathbb{F}$ and $n_1 < n$, applying the induction assumption to the tree $\mathcal{T}$ gives

$$|\Delta_1| \leqslant (n_1+\ell-1)u\delta_1 \quad \text{for} \quad \delta_1 = |\rho| + \sum_{i\in I_1'} |x_i|. \tag{3.4a}$$

On the other hand, applying (2.4) to the sum of reals $\sum_{i\in I_2} x_i$ gives

$$|\Delta_2| \leqslant n_2 u\delta_2 \quad \text{for} \quad \delta_2 = \sum_{i\in I_2} |x_i|. \tag{3.4b}$$

Third, we can bound $|\widehat{s}_1 - s_1|$ as follows. Applying (2.2) to s_1 in (3.1) gives $|\widehat{s}_1 - s_1| \leqslant \min\{u|\widehat{s}_1|, |\widehat{s}_2|\}$. Furthermore, the definitions of the Δ_i and δ_i in (3.2) and (3.4a-b) imply that $|\widehat{s}_i| \leqslant |\Delta_i| + \delta_i$ for $i = 1, 2$. Hence

$$|\widehat{s}_1 - s_1| \leqslant \min\big\{u|\Delta_1| + u\delta_1, |\Delta_2| + \delta_2\big\}. \tag{3.4c}$$

From (3.3) and (3.4) and depending on the expression chosen to bound $|\widehat{s}_1 - s_1|$, we deduce the following two bounds on $|\Delta|$:

$$|\Delta| \leqslant (n_1 + \ell)u\delta_1 + (n_1 + \ell - 1)u^2\delta_1 + n_2 u\delta_2 \tag{3.5}$$

and

$$|\Delta| \leqslant (n_1 + \ell - 1)u\delta_1 + 2n_2 u\delta_2 + \delta_2. \tag{3.6}$$

Recall that $\ell \geqslant 0$, $n_1, n_2 \geqslant 1$, and $n = n_1 + n_2$. Due to the special shape of its $\mathcal{O}(u^2)$-term, the bound in (3.5) implies the desired bound $B := (n_1 + n_2 + \ell - 1)u(\delta_1 + \delta_2)$ on $|\Delta|$ when $u\delta_1 \leqslant \delta_2$ or $(n_1 + \ell - 1)u \leqslant n_2 - 1$. In the remaining case where

$$\delta_2 < u\delta_1 \qquad \text{and} \qquad n_2 - 1 < (n_1 + \ell - 1)u,$$

we prove the bound B by using (3.6). Recalling that $u \leqslant 1$, we have $n_2 - 1 < n_1 + \ell - 1$. This inequality is strict and involves only integers, so it is equivalent to $n_2 \leqslant n_1 + \ell - 1$. Consequently, (3.6) leads to $|\Delta| < (n_1 + \ell)u\delta_1 + (n_1 + n_2 + \ell - 1)u\delta_2 \leqslant B$, as wanted. This completes the proof. ∎

Two observations can be made about Theorem 3.1:

(i) When we say "in the absence of overflow," we mean that overflow occurs neither when rounding the x_i to the $\mathrm{fl}(x_i)$ nor when summing up these rounded values.

(ii) By our definition, underflow occurs if a result is in the subnormal range and causes a rounding error. Thus, in Theorem 3.1 neither the element x_j itself, which is in $\mathbb{F}$, nor the additions can cause underflow, but only the rounding of the reals x_i for $i \neq j$. Assuming that such rounding does not underflow, however, is necessary. To see this we may use arguments similar to the ones given immediately after [5, Proposition 4.1]. For example, consider $x_j = 0$ and for $i \neq j$ let $x_i > 0$ be so small that $\mathrm{fl}(x_i) = 0$. The computed sum $\widehat{s}$ is then equal to zero. Hence, choosing $\rho = \widehat{s} = 0$, we have $|\Delta| = \sum_{i \neq j} |x_i|$, which is generally not upper bounded by $(n + \ell - 1)u|\Delta|$.

Theorem 3.1 provides a general framework capable of handling arbitrary approximations ρ to the computed sum $\widehat{s}$ of the x_i. In the corollary below we specialize this result to the two cases needed to refine the backward error bounds given in [3, Chaps. 8, 9, 10]. The first case is when ρ is the exact product of b and $\mathrm{fl}(\widehat{s}/b)$ for a nonzero floating-point number b, and the second case is when ρ is the exact square of $\mathrm{fl}(\sqrt{\widehat{s}})$.

COROLLARY 3.2. *Let $x_1, \ldots, x_n$ and $\widehat{s}$ be as in Theorem 3.1. Assuming underflow and overflow do not occur, we have the following two error bounds.*

If $b \in \mathbb{F}$ is nonzero and $\widehat{y} = \mathrm{fl}(\widehat{s}/b)$, then

$$|b\widehat{y} - \sum_{i=1}^{n} x_i| \leqslant nu\Big(|b\widehat{y}| + \sum_{i=1, i\neq j}^{n} |x_i|\Big).$$

If $\widehat{s}$ is nonnegative and $\widehat{y} = \mathrm{fl}\big(\sqrt{\widehat{s}}\big)$, then

$$|\widehat{y}^2 - \sum_{i=1}^{n} x_i| \leqslant (n+1)u\Big(\widehat{y}^2 + \sum_{i=1, i\neq j}^{n} |x_i|\Big).$$

Proof. If $\widehat{y} = \mathrm{fl}(\widehat{s}/b)$ without underflow and overflow, then (2.1b) implies $|b\widehat{y} - \widehat{s}| \leqslant u|b\widehat{y}|$, so the first implication follows from applying Theorem 3.1 with $\rho = b\widehat{y}$ and $\ell = 1$. To prove the second implication, it suffices to check that $\widehat{y} = \mathrm{fl}(\sqrt{\widehat{s}})$ implies

$$|\widehat{y}^2 - \widehat{s}| \leqslant 2u\,\widehat{y}^2, \tag{3.7}$$

and then to use Theorem 3.1 with $\rho = \widehat{y}^2$ and $\ell = 2$. To show that (3.7) holds, using only (2.1b) is not enough (see Appendix A) and we therefore resort to the following ufp-based analysis.

Let $y = \sqrt{\widehat{s}}$. Recall from Section 2.1 that neither underflow nor overflow can occur when setting $\widehat{y} = \mathrm{fl}(y)$, and recall from (2.3c) that $\mathrm{ufp}(y) = \widehat{y}$ or $(1 + 2u)\mathrm{ufp}(y) \leqslant \widehat{y}$.

We distinguish two cases. First, if $\mathrm{ufp}(y) = \widehat{y}$ then (2.3a) and (2.1b) imply $\widehat{y} \leqslant y \leqslant (1 + u)\widehat{y}$. Thus, taking squares and using (2.3b), $u \leqslant 1/2$, and the fact that $y^2 = \widehat{s}$ is a floating-point number, we get in this case

$$\widehat{y}^2 \leqslant y^2 \leqslant (1 + 2u)\widehat{y}^2,$$

from which (3.7) follows.

Let us now turn to the case $(1+2u)\mathrm{ufp}(y) \leqslant \widehat{y}$. Rewriting $|\widehat{y}^2 - y^2|$ as $(\widehat{y}+y)|\widehat{y}-y|$ and applying (2.1b) and (2.3d), we obtain in this case

$$|\widehat{y}^2 - y^2| \leqslant (2 + u)\widehat{y} \cdot u\,\mathrm{ufp}(y) \leqslant \tfrac{2+u}{1+2u} u\,\widehat{y}^2 \leqslant 2u\,\widehat{y}^2.$$

Therefore, the bound in (3.7) holds in both cases, which completes the proof. ∎

We conclude this section by noting that if $(\rho, \ell) = (\widehat{s}, 0)$ and all the x_i are in $\mathbb{F}$ and the order of evaluation is fixed to recursive summation, then, in the error bound of Theorem 3.1, both x_1 and x_2 can be omitted and replaced by the computed sum $\widehat{s}$. This may come as a surprise as x_1 and x_2 may be arbitrarily large compared to $\widehat{s}$. However, $|x_1| \gg |\widehat{s}|$ and $|x_2| \gg |\widehat{s}|$ is only possible if x_1 and x_2 are of similar magnitude and opposite signs, that means cancellation occurs in $x_1 + x_2$. But in this case $\mathrm{fl}(x_1 + x_2) = x_1 + x_2$ by Sterbenz' lemma [10], so no rounding error occurs.[2] Note that this argument breaks down if not both x_1 and x_2 are floating-point numbers (take for example $x_1 = -1$ and $x_2 = 1 + u$.) More precisely, the following theorem holds true, the proof of which is deferred to Appendix B.

THEOREM 3.3. *Given $n \in \mathbb{N}_{>0}$, let $x_1, \ldots, x_n$ be in $\mathbb{F}$ and let $\widehat{s}$ be the result of the floating-point evaluation of $x_1 + \cdots + x_n$ by means of recursive summation. Then, in the absence of overflow,*

$$\Big|\widehat{s} - \sum_{i=1}^{n} x_i\Big| \leqslant (n-1)u\Big(|\widehat{s}| + \sum_{i=3}^{n} |x_i|\Big).$$

4. Applications. We show in this section that Theorem 3.1 and Corollary 3.2 can be used to improve upon the following five backward error results from Higham's book [3]: Lemma 8.4, Theorem 8.5 (triangular system solving), Theorem 9.3 (LU factorization), solution to Problem 10.3 (Cholesky variant of Lemma 8.4), and Theorem 10.3 (Cholesky factorization).

Hereafter we write a_{ij} to denote the (i, j) entry of a matrix A, and $\mathrm{diag}(f(i))$ for the $n \times n$ diagonal matrix whose (i, i) entry equals $f(i)$. Hence, whenever we use the diag notation there is an implicit assumption that the index i ranges from 1 to n.

[2] More precisely, in the absence of overflow, $\mathrm{fl}(a-b) = a-b$ whenever $a, b \in \mathbb{F}$ satisfy $a/2 \leqslant b \leqslant 2a$; see [10, p. 138] and [3, p. 45].

4.1. An improved version of Higham's Lemma 8.4. The results of Section 3 imply first that Higham's Lemma 8.4 [3, p. 142] can be rewritten as follows, with the original γ_k and γ_{k-1} terms replaced by ku and $(k-1)u$ and with no restriction on k.

LEMMA 4.1. *Let $k \in \mathbb{N}_{>0}$ and $a_1, \ldots, a_{k-1}, b_1, \ldots, b_{k-1}, b_k, c \in \mathbb{F}$ be given, with b_k nonzero. If $y = \big(c - \sum_{i=1}^{k-1} a_i b_i\big)/b_k$ is evaluated in floating-point arithmetic then, in the absence of underflow and overflow and no matter what the order of evaluation, the computed $\widehat{y}$ satisfies*

$$b_k \widehat{y}(1+\theta_k^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_k^{(i)}), \qquad |\theta_k^{(i)}| \leqslant ku \text{ for all } i.$$

If $b_k = 1$, so that there is no division, then $|\theta_k^{(i)}| \leqslant (k-1)u$ for all i.

Proof. Assume first that $b_k = 1$. In this case, by applying Theorem 3.1 with $\rho = \widehat{y}$, $\ell = 0$, $n = k$, $x_1 = c$ and $x_{i+1} = -a_i b_i$ for $1 \leqslant i < n$, we obtain

$$|\widehat{y} - y| \leqslant (k-1)u\Big(|\widehat{y}| + \sum_{i=1}^{k-1} |a_i b_i|\Big).$$

Hence $\widehat{y} - y = \epsilon\big(|\widehat{y}| + \sum_{i=1}^{k-1} |a_i b_i|\big)$ for some ϵ in $\mathbb{R}$ with $|\epsilon| \leqslant (k-1)u$. Defining $\epsilon^{(0)} = -\mathrm{sign}(\widehat{y})\epsilon$ and $\epsilon^{(i)} = -\mathrm{sign}(a_i b_i)\epsilon$ for $1 \leqslant i < k$, we deduce that $\widehat{y}(1+\epsilon^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\epsilon^{(i)})$, where $|\epsilon^{(i)}| = |\epsilon| \leqslant (k-1)u$ for all i.

The general situation where b_k is a nonzero floating-point number is handled analogously by using Corollary 3.2 with $b = b_k$, $n = k$, and the x_i as above. ∎

Lemma 4.1 then leads immediately to the following backward error result for triangular system solving and improves upon [3, Theorem 8.5]: given $b \in \mathbb{F}^n$ and $T \in \mathbb{F}^{n\times n}$ triangular and nonsingular, then, in the absence of underflow or overflow, substitution produces an approximate solution $\widehat{x}$ to $Tx = b$ that satisfies

$$(T + \Delta T)\widehat{x} = b, \qquad |\Delta T| \leqslant \mathrm{diag}(d_k)|T| \leqslant nu|T| \tag{4.1a}$$

with

$$d_k = \begin{cases} ku, & \text{if } T \text{ is lower triangular,} \\ (n-k+1)u, & \text{if } T \text{ is upper triangular.} \end{cases} \tag{4.1b}$$

If in addition T is *unit* triangular, then d_k can be decreased further to $(k-1)u$ or $(n-k)u$, and the constant nu in the bound (4.1a) can be replaced by $(n-1)u$.

4.2. An improved backward error bound for LU factorization. We now turn to the computation of an LU factorization by means of any variant of Gaussian elimination, and give the following improvement to [3, Theorem 9.3].

THEOREM 4.2. *Let $A \in \mathbb{F}^{m\times n}$ with $m \geqslant n$. If Gaussian elimination runs to completion then, in the absence of underflow and overflow, the computed factors $\widehat{L} \in \mathbb{F}^{m\times n}$ and $\widehat{U} \in \mathbb{F}^{n\times n}$ satisfy*

$$\widehat{L}\widehat{U} = A + \Delta A, \qquad |\Delta A| \leqslant nu|\widehat{L}||\widehat{U}|.$$

If $m = n$ then sharper bounds are

$$\begin{aligned} |\Delta A| &\leqslant \mathrm{diag}\big((i-1)u\big)|\widehat{L}||\widehat{U}| \\ &\leqslant (n-1)u|\widehat{L}||\widehat{U}|. \end{aligned} \tag{4.2}$$

Proof. As shown by Higham in [3, pp. 162–163] it suffices to analyze one of the mathematically equivalent formulations of Gaussian elimination, for example, Doolittle's method: for $k = 1, \dots, n$ and given the first $k-1$ columns of $\widehat{L}$ and the first $k-1$ rows of $\widehat{U}$, the kth row of $\widehat{U}$ and the kth column of $\widehat{L}$ are obtained by floating-point evaluation of the expressions

$$y_{kj} = a_{kj} - \sum_{i=1}^{k-1} \widehat{\ell}_{ki}\widehat{u}_{ij}, \qquad j = k, \dots, n,$$

$$y_{ik} = \Big(a_{ik} - \sum_{j=1}^{k-1} \widehat{\ell}_{ij}\widehat{u}_{jk}\Big)/\widehat{u}_{kk}, \qquad i = k+1, \dots, m,$$

with $\widehat{u}_{kk}$ denoting the computed value of y_{kk}. Writing $\widehat{u}_{kj}$ and $\widehat{\ell}_{ik}$ for the computed values of y_{kj} and y_{ik} for $i, j > k$ and setting $\widehat{\ell}_{kk} = 1$, we deduce from Lemma 4.1 that, no matter what the order of evaluation,

$$\Big|a_{kj} - \sum_{i=1}^{k} \widehat{\ell}_{ki}\widehat{u}_{ij}\Big| \leqslant (k-1)u \sum_{i=1}^{k} |\widehat{\ell}_{ki}||\widehat{u}_{ij}|, \qquad j = k, \dots, n, \tag{4.3a}$$

$$\Big|a_{ik} - \sum_{j=1}^{k} \widehat{\ell}_{ij}\widehat{u}_{jk}\Big| \leqslant ku \sum_{j=1}^{k} |\widehat{\ell}_{ij}||\widehat{u}_{jk}|, \qquad i = k+1, \dots, m. \tag{4.3b}$$

Since $k \leqslant n$, the inequalities in (4.3) imply $|A - \widehat{L}\widehat{U}| \leqslant nu|\widehat{L}||\widehat{U}|$. If in addition $m = n$ then the constant term ku in (4.3b) satisfies $ku \leqslant (i-1)u \leqslant (n-1)u$, which leads to the improved bounds $\operatorname{diag}\big((i-1)u\big)|\widehat{L}||\widehat{U}|$ and $(n-1)u|\widehat{L}||\widehat{U}|$. ∎

4.3. An improved backward error bound for Cholesky factorization. If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, its Cholesky factor is the unique upper triangular matrix $R \in \mathbb{R}^{n\times n}$ such that $A = R^T R$, and by *Cholesky factorization* we mean the computation of R using the conventional algorithm, described for example in [3, Algorithm 10.2]. This algorithm requires to evaluate not only expressions of the form s/b_k with $s = c - \sum_{i=1}^{k-1} a_i b_i$, but also expressions of the form $\sqrt{s}$. Thus we start with the following variant of Lemma 4.1.

LEMMA 4.3. *Let $k \in \mathbb{N}_{>0}$ and $a_1, \dots, a_{k-1}, b_1, \dots, b_{k-1}, c \in \mathbb{F}$ be given, let $s = c - \sum_{i=1}^{k-1} a_i b_i$ be evaluated in floating-point arithmetic, and let $\widehat{s}$ denote the resulting approximation of s. In the absence of underflow and overflow and if $\widehat{s}$ is nonnegative, then, no matter in what order of evaluation $\widehat{s}$ was produced, $\widehat{y} = \mathrm{fl}(\sqrt{\widehat{s}})$ satisfies*

$$\widehat{y}^2(1 + \theta_{k+1}^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i (1 + \theta_{k+1}^{(i)}), \qquad |\theta_{k+1}^{(i)}| \leqslant (k+1)u \text{ for all } i.$$

Proof. This result follows from the second bound in Corollary 3.2 with $n = k$, $x_1 = c$, and $x_{i+1} = -a_i b_i$ for $1 \leqslant i < n$. ∎

Combining Lemmas 4.1 and 4.3 leads to the following backward error bound for the Cholesky factor, which improves upon [3, Theorem 10.3].

THEOREM 4.4. *Let $A \in \mathbb{F}^{n\times n}$ be symmetric. If Cholesky factorization runs to completion then, in the absence of underflow and overflow, the computed factor*

$\widehat{R} \in \mathbb{F}^{n\times n}$ *satisfies*

$$\widehat{R}^T\widehat{R} = A + \Delta A, \qquad |\Delta A| \leqslant \operatorname{diag}\big((i+1)u\big)|\widehat{R}^T||\widehat{R}| \leqslant (n+1)u|\widehat{R}^T||\widehat{R}|. \tag{4.4}$$

Proof. The conventional method for Cholesky factorization computes $\widehat{R}$ one column at a time (cf. Algorithm 10.2 in [3]): for $j = 1, \ldots, n$ and given the first $j-1$ columns of $\widehat{R}$, the jth column is obtained by evaluating in floating-point the expressions

$$y_{ij} = \Big(a_{ij} - \sum_{k=1}^{i-1} \widehat{r}_{ki}\widehat{r}_{kj}\Big)/\widehat{r}_{ii}, \qquad i = 1, \ldots, j-1,$$
$$y_{jj} = \Big(a_{jj} - \sum_{k=1}^{j-1} \widehat{r}_{kj}^2\Big)^{1/2}.$$

Let us denote by $\widehat{r}_{ij}$ and $\widehat{r}_{jj}$ the computed values of y_{ij} and y_{jj}, respectively, no matter what the order of evaluation. Then, applying Lemma 4.1 and Lemma 4.3,

$$\Big|a_{ij} - \sum_{k=1}^{i} \widehat{r}_{ki}\widehat{r}_{kj}\Big| \leqslant iu \sum_{k=1}^{i} |\widehat{r}_{ki}||\widehat{r}_{kj}|, \qquad i = 1, \ldots, j-1, \tag{4.5a}$$

$$\Big|a_{jj} - \sum_{k=1}^{j} \widehat{r}_{kj}^2\Big| \leqslant (j+1)u \sum_{k=1}^{j} \widehat{r}_{kj}^2. \tag{4.5b}$$

The conclusion follows from $A - \widehat{R}^T\widehat{R}$ and $\widehat{R}^T\widehat{R}$ being symmetric. ∎

5. Concluding remarks. The framework introduced in Theorem 3.1 leads to refined backward error bounds for triangular system solving and, perhaps more importantly, for LU and Cholesky factorizations. Thus, when solving a linear system $Ax = b$ by means of such factorizations, it is natural to ask whether the classical backward error bounds for the computed solution $\widehat{x}$ can be improved as well. The answer is yes, but as we will see now, to a lesser extent. Classically, we have $(A + \Delta A)\widehat{x} = b$ with $|\Delta A| \leqslant \gamma_{3n}|\widehat{L}||\widehat{U}|$ if LU factorization is used, and $|\Delta A| \leqslant \gamma_{3n+1}|\widehat{R}^T||\widehat{R}|$ if Cholesky factorization is used; see Theorems 9.4 and 10.4 in [3]. By applying (4.1), (4.2) and (4.4) directly to the proof of these two theorems, improved bounds are easily obtained:

$$|\Delta A| \leqslant \operatorname{diag}\big((n+2i-2)u + n(i-1)u^2\big)|\widehat{L}||\widehat{U}| \leqslant ((3n-2)u + (n^2-n)u^2)|\widehat{L}||\widehat{U}|$$

in the case of LU factorization, and

$$|\Delta A| \leqslant \operatorname{diag}\big((n+2i+1)u + niu^2\big)|\widehat{R}^T||\widehat{R}| \leqslant ((3n+1)u + n^2u^2)|\widehat{R}^T||\widehat{R}|$$

in the case of Cholesky factorization. These bounds hold for all n and no matter what the order of evaluation; in addition, their constants are always smaller than the classical constants γ_{3n} and γ_{3n+1}. However, in both cases a term quadratic in u

remains, and it is not clear whether it can be removed or not. If this is possible then further techniques than those introduced in this paper might be needed to achieve unconditional error bounds whose constants are $(3n-2)u$ and $(3n+1)u$.

But even if it were possible in the example above, it is important to realize that the terms γ_ℓ cannot generally be replaced by ℓu. Indeed, this already happens in the simple case of *pairwise summation* of floating-point numbers. When adding n floating-point numbers $x_1, \dots, x_n$ the classical analysis bounds the absolute error by $\gamma_\ell \sum_{i=1}^n |x_i|$, where ℓ is the height of the binary tree underlying the evaluation order [1, 3]. For pairwise summation this tree has the minimum possible height, namely, $\ell = \lceil \log_2 n \rceil$, but in this case the actual error can be larger than $\ell u \sum_{i=1}^n |x_i|$. To prove this, it suffices to consider the following construction. Let $n = 2^\ell$, assume $\beta = 2$, and let $x_1, \dots, x_n \in \mathbb{F}$ be defined recursively as $x_1 = 1$ and $x_{n/2+i} = u x_i$ for $i = 1, \dots, n/2$. Then, for rounding to nearest with ties broken "to away," a term strictly larger than ℓu is necessary for large enough ℓ. Similar examples can be found for other tie-breaking rules. Although such counterexamples require a large exponent range and a huge dimension, they illustrate the impossibility, in general, to systematically replace γ_ℓ terms by ℓu.

Appendix A. Proof that (2.1b) does not imply (3.7). Let $\mathbb{F}$ with $\beta = 2$ and unit roundoff $u = 2^{-p} \leqslant 1/2$ be given, and consider the set $\overline{\mathbb{F}}$ obtained from $\mathbb{F}$ by replacing 2 and -2 by $2 + 2u + u^2$ and $-(2 + 2u + u^2)$, respectively. Furthermore, let $\overline{\mathrm{fl}}$ denote a round-to-nearest function from $\mathbb{R}$ to $\overline{\mathbb{F}} \cup \{\pm\infty\}$.

In the set $\mathbb{F}$, the predecessor and successor of 2 are $2 - 2u$ and $2 + 4u$, and u is small enough to ensure $2 + 2u + u^2 \leqslant 2 + 4u$. Therefore, $2 - 2u$ and $2 + 2u + u^2$ are two consecutive elements of $\overline{\mathbb{F}}$, and their midpoint, which lies exactly halfway between them, is

$$\mu = 2 + \tfrac{1}{2}u^2.$$

Given a real number t in the normal range of $\overline{\mathbb{F}}$ (which is identical to the normal range of $\mathbb{F}$), the rounding of t to a nearest element of $\overline{\mathbb{F}}$ is modeled as follows: if $2 - 2u < |t| < 2 + 2u + u^2$ then the error relative to $\overline{\mathrm{fl}}(t)$ satisfies

$$\frac{\left|\overline{\mathrm{fl}}(t) - t\right|}{\left|\overline{\mathrm{fl}}(t)\right|} \leqslant \frac{\mu}{2 - 2u} - 1 = u\,\frac{1 + \frac{1}{4}u}{1 - u} =: \overline{u},$$

and otherwise it is bounded by u. Since $u \leqslant \overline{u}$, it follows that the model (2.1b) holds with $\mathbb{F}$, fl, u replaced by $\overline{\mathbb{F}}$, $\overline{\mathrm{fl}}$, $\overline{u}$.

Now, let $\widehat{s} = 4$ and $\widehat{y} = \overline{\mathrm{fl}}(\sqrt{\widehat{s}})$. We see that $\widehat{s}$ belongs to $\overline{\mathbb{F}}$ and, since $2 - 2u \leqslant 2 < \mu$, we have $\widehat{y} = 2 - 2u$. (Note that this is true no matter to which of its neighbors the midpoint μ is rounded.) Hence, on the one hand,

$$|\widehat{y}^2 - \widehat{s}| = 8u - 4u^2$$

and, on the other hand,

$$2\overline{u}\,\widehat{y}^2 = 8u - 6u^2 - 2u^3.$$

Consequently, $|\widehat{y}^2 - \widehat{s}| > 2\overline{u}\,\widehat{y}^2$, that is, (3.7) does not hold for $\overline{\mathbb{F}}$, $\overline{\mathrm{fl}}$, $\overline{u}$. This concludes the proof that assuming only (2.1b) is not enough to ensure (3.7). ∎

Appendix B. Proof of Theorem 3.3. The proof is by induction on n and borrows some ideas from [7]. The case $n = 1$ is trivial and the case $n = 2$ corresponds

to the second standard model, given in (2.1b). Let us now assume that $n \geqslant 3$ and that the result is true up to $n-1$. Writing $\widehat{s}_1 = x_1$, we start by defining for $2 \leqslant i \leqslant n$

$$s_i = \widehat{s}_{i-1} + x_i, \qquad \widehat{s}_i = \mathrm{fl}(s_i), \qquad \Delta_i = \widehat{s}_i - (x_1 + \cdots + x_i). \tag{B.1}$$

With this notation our goal is to show that $|\Delta_n| \leqslant (n-1)u\big(|\widehat{s}_n| + \sum_{i=3}^{n} |x_i|\big)$.

It is easily verified from (B.1) that for $2 \leqslant j < n$ the following relations hold:

$$\Delta_n = \sum_{i=j+1}^{n} (\widehat{s}_i - s_i) + \Delta_j \qquad \text{and} \qquad \widehat{s}_j = \sum_{i=j+1}^{n} (s_i - \widehat{s}_i - x_i) + \widehat{s}_n. \tag{B.2}$$

Let us first use (B.2) with $j = n-1$. The identity on the left gives

$$\begin{aligned} |\Delta_n| &\leqslant |\widehat{s}_n - s_n| + |\Delta_{n-1}| \\ &\leqslant u|\widehat{s}_n| + (n-2)u\Big(|\widehat{s}_{n-1}| + \sum_{i=3}^{n-1} |x_i|\Big), \end{aligned}$$

by using (2.1b) together with the induction hypothesis. Since the right identity in (B.2) and (2.1b) also lead to $|\widehat{s}_{n-1}| \leqslant |\widehat{s}_n - s_n| + |x_n| + |\widehat{s}_n| \leqslant (1+u)|\widehat{s}_n| + |x_n|$, we arrive at

$$|\Delta_n| \leqslant (n-1)u|\widehat{s}_n| + (n-2)u^2|\widehat{s}_n| + (n-2)u\sum_{i=3}^{n} |x_i|. \tag{B.3}$$

Let us now use (B.2) with $j = 2$. On the one hand,

$$\begin{aligned} |\Delta_n| &\leqslant \sum_{i=3}^{n} |\widehat{s}_i - s_i| + |\Delta_2| \\ &\leqslant \sum_{i=3}^{n} |x_i| + u|\widehat{s}_2|, \end{aligned} \tag{B.4}$$

since $|\widehat{s}_i - s_i| \leqslant |x_i|$ by (B.1) and (2.2) and since $|\Delta_2| = |\widehat{s}_2 - s_2| \leqslant u|\widehat{s}_2|$ by (2.1b). On the other hand,

$$\begin{aligned} |\widehat{s}_2| &\leqslant \sum_{i=3}^{n} |s_i - \widehat{s}_i| + \sum_{i=3}^{n} |x_i| + |\widehat{s}_n| \\ &\leqslant 2\sum_{i=3}^{n} |x_i| + |\widehat{s}_n|. \end{aligned} \tag{B.5}$$

Hence, from (B.4) and (B.5) and since by assumption $2 \leqslant n-1$,

$$|\Delta_n| \leqslant u|\widehat{s}_n| + \sum_{i=3}^{n} |x_i| + (n-1)u\sum_{i=3}^{n} |x_i|. \tag{B.6}$$

The desired bound $|\Delta_n| \leqslant (n-1)u\big(|\widehat{s}_n| + \sum_{i=3}^{n} |x_i|\big)$ then follows immediately from either (B.3) or (B.6), depending on how $(n-2)u|\widehat{s}_n|$ compares with $\sum_{i=3}^{n} |x_i|$. This concludes the proof of Theorem 3.3. ∎

Acknowledgments. We thank the editor and two anonymous referees for their thorough reading and helpful suggestions to improve in particular the rationale of the paper.

REFERENCES

[1] T. O. Espelid, *On floating-point summation*, SIAM Rev., 37 (1995), pp. 603–607.
[2] J. R. Hauser, *Handling floating-point exceptions in numeric programs*, ACM Trans. Program. Lang. Syst., 18 (1996), pp. 139–174.
[3] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, second ed., 2002.
[4] IEEE Computer Society, *IEEE Standard for Floating-Point Arithmetic*, IEEE Standard 754-2008, Aug. 2008. available at `http://ieeexplore.ieee.org/servlet/opac?punumber=4610933`.
[5] C.-P. Jeannerod and S. M. Rump, *Improved error bounds for inner products in floating-point arithmetic*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 338–344.
[6] W. Kahan, *A brief tutorial on gradual underflow*, 2005. Available at `http://www.cs.berkeley.edu/~wkahan/`.
[7] S. M. Rump, *Error estimation of floating-point summation and dot product*, BIT, 52 (2012), pp. 201–220.
[8] S. M. Rump, T. Ogita, and S. Oishi, *Accurate floating-point summation, Part I: Faithful rounding*, SIAM J. Sci. Comput., 31 (2008), pp. 189–224.
[9] J. R. Shewchuk, *Adaptive precision floating-point arithmetic and fast robust geometric predicates*, Discrete and Computational Geometry, 18 (1997), pp. 305–363.
[10] P. H. Sterbenz, *Floating-Point Computation*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1974.
[11] J. H. Wilkinson, *Numerical linear algebra on digital computers*, IMA Bulletin, 10 (1974), pp. 354–356.